

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

(19)



Eur pâis hes Patentamt
European Patent Office
Office européen des brevets



(11)

EP 0694829 A1

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:
31.01.1996 Bulletin 1996/05

(51) Int Cl. 6: G06F 3/023, G06F 17/30

(21) Application number: 95305052.3

(22) Date of filing: 20.07.1995

(84) Designated Contracting States:
DE FR GB

(72) Inventors:
• Bowers, Frank H.
Mountain View, California 94040 (US)
• Card, Stuart K.
Los Altos, California 94022 (US)

(30) Priority: 28.07.1994 US 283004

(74) Representative: Reynolds, Julian David et al
Marlow, Buckinghamshire SL7 1YL (GB)

(54) A method and apparatus for visualization of database search results

(57) A method and apparatus for representing the results of a search of a database. The present invention provides for creating a view of database search results via a tree structure in which detail is selected and context preserved. In the present invention, the tree structure (Fig.2a) is created (402) based on user specified parameters. These parameters represent attributes of documents stored in the database and may differ from the search parameters. The tree structure is then mapped (404) to a static reference surface which is visually perceived as three-dimensional. The reference surface is comprised of a detail area where detail of the tree structure (Fig.2b) is displayed and a context area for displaying other portions of the tree in less detail but which conveys to the viewer a sense of context. The tree structure may be scrolled about the reference surface to bring portions of the structure into a direct detail view while retaining a context view of the overall tree.

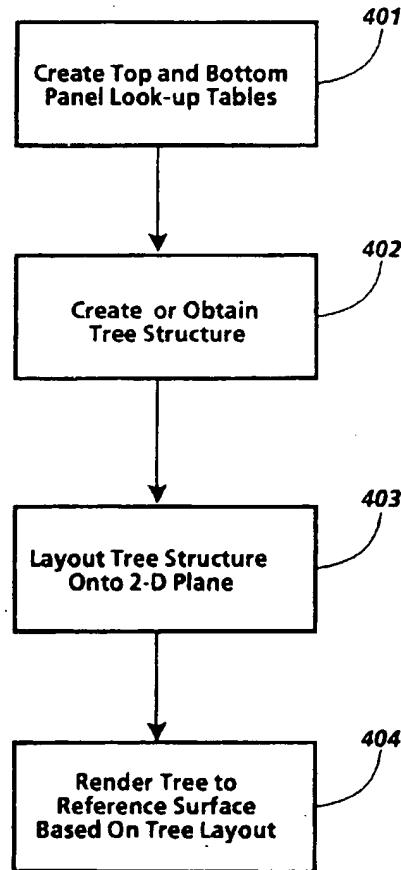


Fig. 4

Description

The present invention relates generally to the field of information display, and in particular to providing a visual display relating to the results of a database search.

It is well understood that information visualizations provide insights to the underlying data. For example, for any given set of information, new information regarding patterns or relationships can be obtained by looking at attributes of the information. It is often useful to view the attributes of the information in a hierarchical structure. However, a difficulty lies in displaying large hierarchical structures in a limited display area.

EP-A-435,601 discloses a technique for displaying and manipulating three-dimensional representations of a tree structure which have rotating substructures.

US-A-4,752,889 describes a graphic display that shows links between chunks of knowledge. The user, with mouse clicks, can obtain a display of links from a displayed chunk of knowledge to other chunks of knowledge, which are added to the display. Screen scrolling mechanisms allow movement from one area of the overall graph to another.

EP-A-535,986 describes a method for centering a selected node of a node link structure along a centering line. The nodes are in rows, and each row extends across a centering line with links between nodes in adjacent rows. When a user requests a centering operation for an indicated node, a sequence of images is presented, each including a row that appears to be a continuation of the row with the indicated node and that includes a continued indicated node that appears to be a continuation of the indicated node.

Furnas, G.W., "Generalized Fisheye Views," CHI '86 Proceedings, ACM, April 1986, pp. 16-23, describes fisheye views that provide a balance of local detail and global context. Section 1 discusses fisheye lenses that show places nearby in great detail while showing the whole world, showing remote regions in successively less detail; a caricature is the poster of the "New Yorker's View of the United States." Section 3 describes a degree of interest (DOI) function that assigns to each point in a structure, a number telling how interested the user is in seeing that point, given the current task. A display can then be made by showing the most interesting points, as indicated by the DOI function. The fisheye view can achieve, for example, a logarithmically compressed display of a tree, as illustrated by Fig. 4 of Furnas for a tree structured text file. Section 4 also describes fisheye views for botanical taxonomies, legal codes, text outlines, a decisions tree, a telephone area code directory, a corporate directory, and UNIX file hierarchy listings. Section 5 indicates that a display-relevant notion of a priori importance can be defined for lists, trees, acyclic directed graphs, general graphs, and Euclidean spaces, unlike the geographical example which inspired the metaphor of the "New Yorker's View," the underlying structures need not be spatial, nor need the output be graphic.

Fig. 6 of Furnas shows a fisheye calendar.

EP-A-447,095 discloses a processor which presents a sequence of images of a workspace that is stretched to enable the user to view a part of a workspace in greater detail. The workspace includes a middle section and two peripheral sections that meet the middle section on opposite edges. Each of the sections appears to be a rectangular two-dimensional surface and they are perceptible in three dimensions. When the user is viewing the middle section as if it were parallel to the display screen surface, each peripheral section appears to extend away from the user at an angle from the edge of the middle section so that the peripheral sections occupy relatively little of the screen. When the user requests stretching, the middle section is stretched and the peripheral sections are compressed to accommodate the stretching. When the user requests destretching, the middle section is destretched and the peripheral sections are decompressed accordingly.

The present invention seeks to provide for the organization and visualization of database search results independent of the search parameters. Often, it is desirable to obtain properties of the search results independent of the search parameters. For example, a search on a database may be performed by specifying the authors. It may be undesirable to limit the search by specifying a date. It may later be desirable to organize the search results by date in order to determine when particular documents were created.

The present invention provides a method for generating hierarchically related information for the results of a query to a database, said database for storing documents, said method comprising the steps of:

35 a) executing a search on a database to obtain a set of documents;

40 b) receiving a set of N ordered user specified view preferences, each of said user specified view preferences defining a property of information stored in said database;

45 c) generating an N + 1 level tree structure based on said set of documents and said set of N ordered user specified view preferences;

50 d) displaying a reference surface on which said tree structure is displayed, said reference surface having a first area for showing a first portion of the tree structure in detail and a second area for showing second portions of said tree structure lacking detail;

55 e) determining a layout of said tree structure relative to said reference surface;

f) mapping said tree structure to said reference surface based on said layout to create tree display data; and

g) displaying said tree structure on said reference surface.

The invention further provides a computer controlled display system according to claim 5 of the appended claims.

In the present invention, a tree structure is created based on the search results and user specified parameters. The user specified parameters represent attributes of documents stored in the database and may differ from the search parameters. The tree structure is then mapped to a static reference surface which is visually perceived as three-dimensional. The reference surface is comprised of a detail area where detail of the tree structure is displayed and a context area for displaying other portions of the tree in less detail but which conveys to the viewer a sense of context. The tree structure may be scrolled about the reference surface to bring portions of the structure into a direct detail view while retaining a context view of the overall tree.

Figure 1 is a block diagram of computer controlled display system in the currently preferred embodiment of the present invention.

Figure 2a illustrates a tree structure laid out on a plane.

Figure 2b illustrates the tree structure of Figure 2a which has been mapped to a reference surface of a preferred embodiment of the present invention.

Figure 3a illustrates a reference surface which is perceived as three-dimensional as is utilized in a preferred embodiment of the present invention.

Figure 3b is a side view of the reference surface of Figure 3a with respect to a user viewpoint.

Figure 4 is a flowchart of the general steps performed when creating the visualization in a tree structure in a preferred embodiment of the present invention.

Figure 5 is an example of a tree structure showing the results of a database search organized based on user specified preferences.

Figure 6 is a flowchart of user interactions that may be performed in a preferred embodiment of the present invention.

Figure 7 is a flowchart of the steps performed for generating the visualization of a tree structure in a preferred embodiment of the present invention.

Figure 8 is a flowchart of the steps for creation of the tree structure in a preferred embodiment of the present invention.

Figure 9 is a flowchart of the steps for laying out a tree structure onto a plane in a preferred embodiment of the present invention.

Figure 10 is a flowchart of the steps for scrolling the tree structure about the reference surface as may be performed in a preferred embodiment of the present invention.

Figure 11 illustrates a data structure for a node in a preferred embodiment of the present invention.

Figure 12 a screen display of a tree view of the re-

sults of a database search in a preferred embodiment of the present invention.

Figure 13 is a screen display showing a highlighted path, in a preferred embodiment of the present invention.

Figure 14 is a screen display showing a highlighted path after it has been scrolled into view, in a preferred embodiment of the present invention.

A method and apparatus for visualization of the results of a database search is described herein. In the following description numerous specific details are set forth, such as data models for representing a tree structure, in order to provide a thorough understanding of the present invention. It would be apparent, however, to one skilled in the art to practice the invention without such specific details. In other instances, specific implementation details such as the system calls for causing a tree structure to be displayed on a screen, have not been shown in detail in order not to unnecessarily obscure the present invention.

Hierarchically related information is often represented as a tree. As used herein, the term node will refer to a point on the tree structure. Each node includes pointers to a parent node, a child node and a sibling node to create the tree structure. A root node is the highest level node. Leaf nodes are the bottom level nodes.

When a tree structure becomes large, it may be difficult to present the entire tree on a display area in full detail. Thus, it may be difficult to trace that path of a node. Further, it is often desirable to know where a particular node in a tree structure is located in order to understand the context of the data. The present invention addresses this by providing a three-dimensional context view of the tree which can be scrolled.

It should be noted that a tree is in the general class of node-link structures. As will become apparent in the description below, the present invention could be implemented so as to support other node link structures.

Various techniques are known in the art for rendering three-dimensional images on two dimensional display screens. However, manipulating three-dimensional images is computationally intensive. Often, expensive 3-D graphics hardware is used in order to render images in an acceptable timeframe. The present invention avoids the need for using sophisticated hardware to provide a three dimensional visualization which can be manipulated. In the present invention, a three dimensional reference surface is created, about which the information moves. This provides for seeing the entire composition or path to the end nodes of a tree structure.

As will become apparent in the description below, a preferred embodiment of the present invention provides visualizations of the results of a database search (often, important information can be determined by examining relationships between the various search results.) However, it would be apparent to one of skill in the art to apply the present invention to other hierarchically structured information.

Overview of a Computer Controlled Display System in the Currently Preferred Embodiment of the Present Invention

Referring to Figure 1, the computer based system on which the currently preferred embodiment of the present invention may be implemented is comprised of a plurality of components coupled via a bus 101. The bus 101 may consist of a plurality of parallel buses (e.g. address, data and status buses) as well as a hierarchy of buses (e.g. a processor bus, a local bus and an I/O bus). In any event, the computer system is further comprised of a processor 102 for executing instructions provided via bus 101 from internal memory 103 (note that the internal memory 103 is typically a combination of Random Access and Read Only Memories). The processor 102 will be used to perform various operations in support of creating the tree visualizations. Such operations that would be performed by the processor 102 are described with reference to Figures 6-10. The processor 102 and internal memory 103 may be discrete components or a single integrated device such as an Application Specification Integrated Circuit (ASIC) chip.

Also coupled to the bus 101 are a keyboard 104 for entering alphanumeric input, external storage 105 for storing data, a cursor control device 106 for manipulating a cursor, and a display 107 for displaying visual output. The keyboard 104 would typically be a standard QWERTY keyboard but may also be telephone like keypad. The external storage 105 may be fixed or removable magnetic or optical disk drive. The cursor control device 106, e.g. a mouse or trackball, will typically have a button or switch associated with it to which the performance of certain functions can be programmed.

The currently preferred embodiment of the present invention is designed for use on a commercially available IBM compatible computer system available from various vendors, and which is running the Microsoft Windows 3.1 Operating System. The Windows operating system provides for the use of multiple workspaces concurrently. The operation of Windows-like operating systems is well known in the art.

In operation, the currently preferred embodiment of the present invention will also make use of various user interface facilities. The cursor control device is used to position a cursor located on the display to a desired location (i.e. "point" to the desired location). When the cursor control device is positioned (i.e. pointing) to something on a display, the button associated with the cursor control device may be depressed and released in order to invoke an action. This is referred to as a "click". A different function may be invoked if the button is depressed and released twice in succession. This is referred to as a "double click." When a user points to an object on the display, depresses the button and holds it down, the object will follow the cursor and be placed at the location on the display where the button is released. This is referred to as a "drag and drop" or "drag" operation.

Overview of the Tree Visualization

Figures 2a and 2b illustrate the visualization of a tree structure in the currently preferred embodiment. Figure 2a illustrates a tree structure 201 which is projected onto a plane 202. Referring to Figure 2b, the same tree structure 201 has been mapped onto a reference surface 203. The reference surface 203 has a center panel 204 sharing an edge with a top panel 205 and a bottom panel 206. The center panel 204 has a rectangular shape while the top panel 205 and bottom panel 206 have trapezoidal shapes.

Although not illustrated, the reference surface may be rotated by 180 degrees. This may be done if it is desirable to have a vertical representation of a tree structure. Further, the various levels of the tree structure of Figures 2a and 2b are aligned in columns. As will become apparent in the description below, such an alignment reduces processing computations for scrolling of the tree structure across the reference surface and does not effect the underlying nature of the tree structure. However, it would be apparent to one of skill in the art to practice the present invention with a tree structure where the various levels are not vertically aligned in columns

The reference surface provides visual cues so that the visualization is perceived as three dimensional. Figures 3a and 3b illustrate the reference surface in greater detail. Figure 3a illustrates a reference surface as displayed. The reference surface 300 is comprised of a center panel 301, a top panel 302 and a bottom panel 303. The top panel 302 is perceived as folded back from center panel 301 at fold edge 304. The bottom panel 303 is perceived as folded back from center panel 301 at fold edge 305. In order to provide visual cues as to three-dimensions (i.e. it is folded back) the top panel 302 and bottom panel 303 have different fill patterns than center panel 301. Further, the background of the reference surface may also be shaded to provide the three-dimensional visual effect.

Figure 3b is a side view of the reference surface 300 with respect to user viewpoint 306. From Figure 3b, it is readily observed that the user viewpoint 306 is orthogonal to center panel 301, whereas information on top panel 302 and bottom panel 303 are not. Accordingly, information in top panel 302 or bottom panel 303 are viewed in perspective.

It should further be noted that the size of the display area for the reference surface may change. In the currently preferred embodiment when the width of the display area changes, all the panel widths correspondingly change. However, if there is a change to the height of the display area, only the center panel will change height. This is done to show the maximum number of nodes in detail while retaining context. And as will be described in greater detail below, it negates the need to update look-up tables used in creating the three-dimensional effect of the tree structure.

Figure 4 is a flowchart describing the general steps

performed for the rendering of the present invention's visualization of a tree structure. First, perspective transformation look-up tables for the top and bottom panels are created, step 401. These look-up tables are for creating the display data for portions of the tree structure on the top and bottom panels. The look-up tables are static for the durations of the visualization. In a preferred embodiment, each look-up table stores 3000 sample points for transforming an orthogonal view to a perspective view. Values between the sample points are obtained through interpolation. Techniques for creating such a look-up table to create a perspective view are well known in the art. Thus, further description on the generation of sample points in the look-up table is not deemed necessary. Next, the tree structure (i.e. the node/link pointer data) is created or otherwise obtained, step 402. Creation of the tree structure is dependent on what the structure represents. In a preferred embodiment, a hierarchical view of the results of a database search are embodied in the tree structure. A description of the creation of the tree structure in a preferred embodiment is provided below. Once the tree structure is created or obtained, it is mapped to the reference surface by the following steps. First, the tree structure is laid out on a two dimensional plane, step 403. Initial layout of the tree structure will center the tree on the reference surface. Subsequent layouts, e.g. as an effect of scrolling, will cause the layout positions to move. In the laying out step, each node will obtain a planar address. The tree is then rendered onto the reference surface, step 404. This is accomplished by transforming the plane coordinates for the nodes to screen coordinates using the look-up tables generated in step 401 for nodes in the top and bottom panels and direct scaling for nodes in the center panel.

When the visualization is presented to a user, the reference surface is displayed and the tree structure overlaid. The user invocation of and processing for scrolling of the tree structure is described in greater detail below.

The rendering of the visualization is less computationally intensive because of the assumptions that the user's point of view is constant, i.e. orthogonal to the center panel and that the top and bottom panels are symmetric. However, it retains the benefit of a three-dimensional representation of viewing both context and detail concurrently.

The present invention is now described in reference to a preferred embodiment of viewing the results of a database search. However, it would be apparent to one of skilled in the art that the visualization of the present invention could be used for viewing any information capable of representation in a tree structure.

Information Visualization of Database Search Results

In a preferred embodiment of the present invention, a database maintains a collection of documents. Each document has multiple indices by which it can be refer-

enced. The present invention allows the results of a database search to be organized and viewed in a manner based on these indices. The indices used for organization and viewing may be different from the indices used for the database search. The indices which a user specifies by which they would like to view the search results are termed preferences.

The preferences that a user specifies are ordered. A first preference is used to create the top level of the tree structure, a second preference the second level, and so on until the bottom level of the structure which are representations of the individual items of the search results (i.e. the documents.) Typically, the tree structure created from the search results will be shallow (the number of specified preferences plus 1) and broad (the number of documents in the search result.) A label on each node will indicate the corresponding value for the preference. Figure 5 illustrates an example of a view of a search result. In this view, a first preference is Authors, a second preference is Type and a third preference is Date. Referring to Figure 5, a top level 500 identifies three authors, Allen (node 501), Bowers (node 502) and Card (node 503). Following the Allen branch, in the second level of the hierarchy, Type, a grouping 504 of document types is specified, namely Spreadsheet (node 504a), Report (node 504b) and Memo (node 504c). From the third level of the hierarchy, it is observed that the spreadsheet is dated 6/15/93 (node 505a), the report dated 12/1/93 (node 505b) and the Memo on 6/30/94 (node 505c). Finally, we see that from the bottom level of the hierarchy the documents associated with the Allen branch (nodes 506-508).

Following the Bowers branch, it is observed from the second level that documents authored by Bowers are of type Graphics (node 509a) or Code (node 509b). The documents of type Graphics were dated on 1/1/93 (node 510a) and the code on 5/15/94 (node 510b). Finally, the documents associated with the Bowers chain are indicated in nodes 511-514.

Following the Card branch, it is observed from the second level that documents authored by Card are of type Audio (node 515a) or Image (node 515b). From the third level, the documents of type Audio are dated 3/24/94 (node 516a) and the documents of type Image are dated 2/16/94 (node 516b). Finally, the documents associated with the Card branch are defined by nodes 517-519.

In a preferred embodiment, each database that may be searched will have default preferences that may be used for viewing a tree structure.

It is also worth noting that the preferred embodiment operates within the parameters of a window system. For example, a display view may only show a portion of a tree structure because the window has been sized to be smaller than the actual size of the window display area.

The preferred embodiment of the present invention is further described with reference to the flowchart of Figure 6. Referring to Figure 6, a user first selects a data-

base and performs a database search, step 601. Once the search is completed, the user may then specify how they would like to view the search results. In a preferred embodiment the user may view the results as a list of documents or in a tree structure. The user selects the tree view, step 602. In the preferred embodiments, the various view options are represented as icons which the user selects by a point and click operation. Other means of view selection, e.g. via a pull-down menu could be used. Once the user selects the tree view, they must specify the preferences upon which to build the tree structure, step 603. In a preferred embodiment, a user may select up to 5 preferences. The primary limitation being the size of a typical display. It would be apparent to one skilled in the art to allow a fewer or greater number of preferences.

The tree structure for the view is then created and displayed based on the search results and the preferences, step 604. The manner in which the tree structure is built is described in greater detail below. At this point, the user may highlight a document path, view a document or cause a document to be retrieved from the database. It should be noted that in a preferred embodiment of the present invention, each node is capable of being selected by a point operation using the cursor control device. For any operation, the user first points to a desired node, step 605. Highlighting the document path through the tree structure is invoked by a single click of the cursor control device. The highlighting is accomplished by displaying the path in reverse video (see Figure 12). Before highlighting it must be determined if the document node corresponding to the pointed to node is in the center panel, step 606. If the document node is in the center panel, the document path is highlighted, step 608. If the document node is not in the center panel, the tree view is rotated (i.e. scrolled) to bring the document node into the center of the center panel, step 607. This scrolling is done automatically. When completed, the document node will be centered in the center panel and the path highlighted per step 608.

Viewing a document, step 609, is invoked by a double click of the cursor control device. The viewing mechanisms of a preferred embodiment provide for retrieval of the corresponding document and for presentation on a window in the display. Retrieving a document for editing purposes, step 610 is accomplished through a drag and drop operation (into a predefined work area). Note that in a preferred embodiment, the node pointed to need not be the document node. It could be any node in the tree hierarchy which ultimately point to the desired document.

The creation and rendering of a tree structure in the currently preferred embodiment is described with reference to the flowchart of Figure 7. Referring now to Figure 7, the underlying tree representation is created from the search results and the provided preferences, step 701. In a preferred embodiment, the tree consists of virtual nodes, the inner nodes which correspond to the preference

values, and actual nodes, the leaf nodes (displayed in the right-most column) which correspond to the documents resulting from the search. In a preferred embodiment, the creation of the tree is accomplished through recursive calls to a software program called AddNode. A flowchart outlining the functionality of AddNode is provided in Figure 8. AddNode takes as input, a parent node and a document. In creating the tree AddNode is called for each document, specifying a root node as the parent node. Referring now to Figure 8, the value of the preference, say X, for the current depth in the hierarchy is obtained, step 801. This would involve examining the database to determine the value of the index corresponding to the preference. A search is then made to determine if any child node has the value X, step 802. If no child node has the value X, then a new virtual child node having the value X is created, step 803. The depth of the hierarchy is then incremented, step 804. A check is then made to determine if the maximum depth in the hierarchy has been reached, step 805. If it has, then an actual node is created whose parent is the child with the value X (i.e. the current child), step 806. If a child having value X exists, then AddNode is recursively called with that child as the new parent, step 807.

Referring back to Figure 7, the tree structure is then laid out on a plane, step 702. It should first be noted that in a preferred embodiment each level in the hierarchy is in a column (which is not visible) of the display window. The steps for laying out the tree structure on a plane are illustrated in the flowchart of Figure 9. Referring to Figure 9, the tree is traversed to count the number of nodes at each level in the hierarchy, step 901. Each level will form a column so a node count for each column is created. In a preferred embodiment, each node when displayed is 80 × 20 pixels in size. Thus, the size of a column can be readily determined. This will correspond to the number of nodes in a column. Next, a bounding rectangle for each column is determined, step 902. The bounding rectangle is based on the number of columns, the number of nodes in each column, and the dimensions of the display window. The bounding rectangle will encompass all the nodes in a level of the hierarchy. The tree is then traversed to place the nodes in the plane in terms of plane coordinates, step 903. For the placement, the x coordinate is determined by the node's depth in the hierarchy and the number of levels in the hierarchy. The y coordinate is determined by the node's position within the column as the tree structure is traversed.

Referring back to Figure 7, the tree layout is then rendered on the reference surface, step 701. This is accomplished by transforming the plane coordinates to display screen coordinates. The transformation that takes place depends on the panel. As described above, the reference surface relates to the plane in that it merely folds at predetermined vertical fold locations (i.e. at a particular Y-coordinate in the plane). So determination of which transformation to use requires examination of the Y-coordinate of the node. For the center panel, plane

coordinates are scaled to the those of the reference surface. For the top and bottom panels look-up tables are used to apply perspective transformations on the plane coordinates. For each of the top and bottom panels two look-up tables are used. One look-up table maps plane y-distance from a panel fold to screen y-distance. A second look-up table maps plane x-distance from the center to screen x-distance from the center.

As described above with respect to Figure 6, when a leaf node is in the top or bottom panel and the path to that leaf node is selected, scrolling of the tree may occur. Figure 10 is a flowchart describing the steps for scrolling. The scrolling of a preferred embodiment is intended to provide animation. It has been determined that animation aids the users perception of what is occurring. To accommodate animation, the scrolling occurs in animation steps with a total animation time of 0.6 seconds (which has been determined to be optimal). Referring to Figure 10, First the distance each column of the tree will need to travel is determined, step 1001. In a preferred embodiment the selected nodes, its ancestors and its children will be centered on the reference surface. The distance that must be traveled in each animation step so that the most distant column will get to its destination is then determined, step 1002. This distance is based on the time required for recent tree renderings. This distance will provide an offset for each column for each animation step. The animation steps are then performed, step 1003. Here, tree is rendered on the reference surface incrementing each column's offset towards its destination at each step until all of the columns are at their final destinations.

Internal Representation of a Node

Figure 11 illustrates a data structure for a node in a tree structure in the currently preferred embodiment of the present invention. The structure is comprised of a plurality of pointers 1101-1104 and a layout address for the node, 1105. The pointer 1101 points to a parent node, the pointer 1102 points to a child node, the pointer 1103 points to a sibling node and the pointer 1104 points to document information. The pointers 1101-1103 are typical for a tree structure representation. The pointer 1104 is used to point to the document itself. The layout address 1105 contains the address for the node with respect to the plane as generated in the layout plane step described in Figure 9.

Screen Displays of the Currently Preferred Embodiment.

Figures 12-14 "Copyright 1994, Xerox Corporation" (17 U.S.C. 401) are screen displays illustrating the information visualization of database search. Referring to Figure 12, a tree of a search result is illustrated. Here we see that various nodes have been rendered on the top and bottom panels. Figure 13 illustrates the highlighting

of a node along with its descendant nodes. Referring to Figure 13, the node 1301 is selected and highlighted. This results in the descendant nodes 1302-1305 to also be highlighted. Finally, Figure 14 illustrates a selection which causes scrolling. Assuming the starting position as illustrated in Figure 12, node 1401 is selected. The descendant nodes 1402-1411 are then scrolled to the center panel and highlighted. Note that nodes have scrolled off the center panel and onto the bottom panel.

Claims

1. A method for generating hierarchically related information for the results of a query to a database, said database for storing documents, said method comprising the steps of:

10 a) executing a search on a database to obtain a set of documents;

15 b) receiving a set of N ordered user specified view preferences, each of said user specified view preferences defining a property of information stored in said database;

20 c) generating an N + 1 level tree structure based on said set of documents and said set of N ordered user specified view preferences;

25 d) displaying a reference surface on which said tree structure is displayed, said reference surface having a first area for showing a first portion of the tree structure in detail and a second area for showing second portions of said tree structure lacking detail;

30 e) determining a layout of said tree structure relative to said reference surface;

35 f) mapping said tree structure to said reference surface based on said layout to create tree display data; and

40 g) displaying said tree structure on said reference surface.

45 2. The method as recited in Claim 1 wherein each node in said N + 1 level of said tree structure points to a corresponding document and each of said N levels corresponds to one of said N ordered user specified view preferences.

50 3. The method as recited in Claim 2 wherein said step of generating an N + 1 level tree structure is further comprised of:
55 for each document at each of said 1 to N levels performing the steps of:

c1) retrieving a value for said document at a level being processed;

c2) determining if a child node exists having said value; 5

c3) if no child nodes exists, creating a new child node having said value and processing next level; and 10

c4) if a child exists having said value, processing next level.

4. The method as recited in Claim 1 wherein N is less than or equal to 5. 15

5. A computer controlled display system for displaying the results of queries to a database comprising:
 receiving means for receiving a set of document indicators, said set of document indicators corresponding to the results of a database query; 20
 means for specifying one or more search preferences;
 means for constructing a tree structure based on said one or more search preferences and said set of document indicators; 25
 means for generating a reference surface on which said tree structure is displayed, said reference surface having a first area for showing a first portion of the tree structure in detail and a second area for showing second portions of said tree structure lacking detail; 30
 layout generation means for generating a layout of said tree structure relative to said reference surface; 35
 mapping said tree structure to said reference surface based on said layout to create tree display data; and
 display means for displaying said tree structure on said reference surface. 40

6. The system as recited in Claim 5 further comprising of means for causing said tree structure to be scrolled across said reference surface. 45

7. The system as recited in Claim 5 or 6 further comprising means for highlighting a document path in said tree structure.

8. The system as recited in Claims 5, 6 or 7 further comprising means for enabling viewing a document from said tree structure. 50

9. The system as recited in any of Claims 5 to 8 further comprising means for enabling retrieval of a document for editing from said tree structure. 55

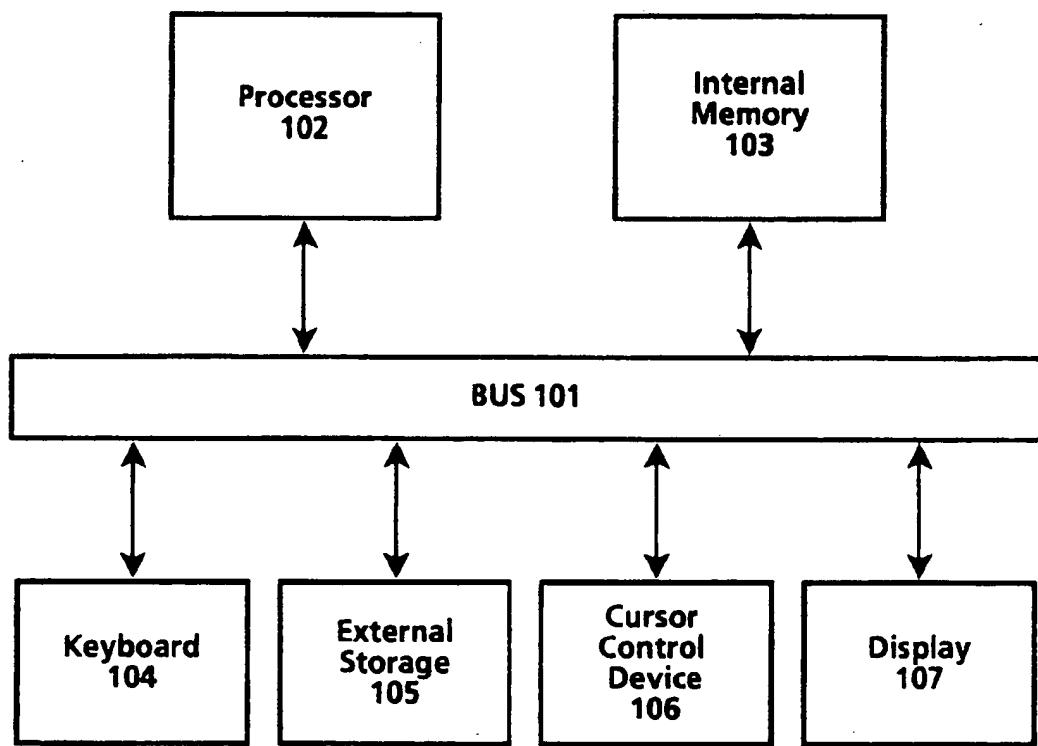


Fig. 1

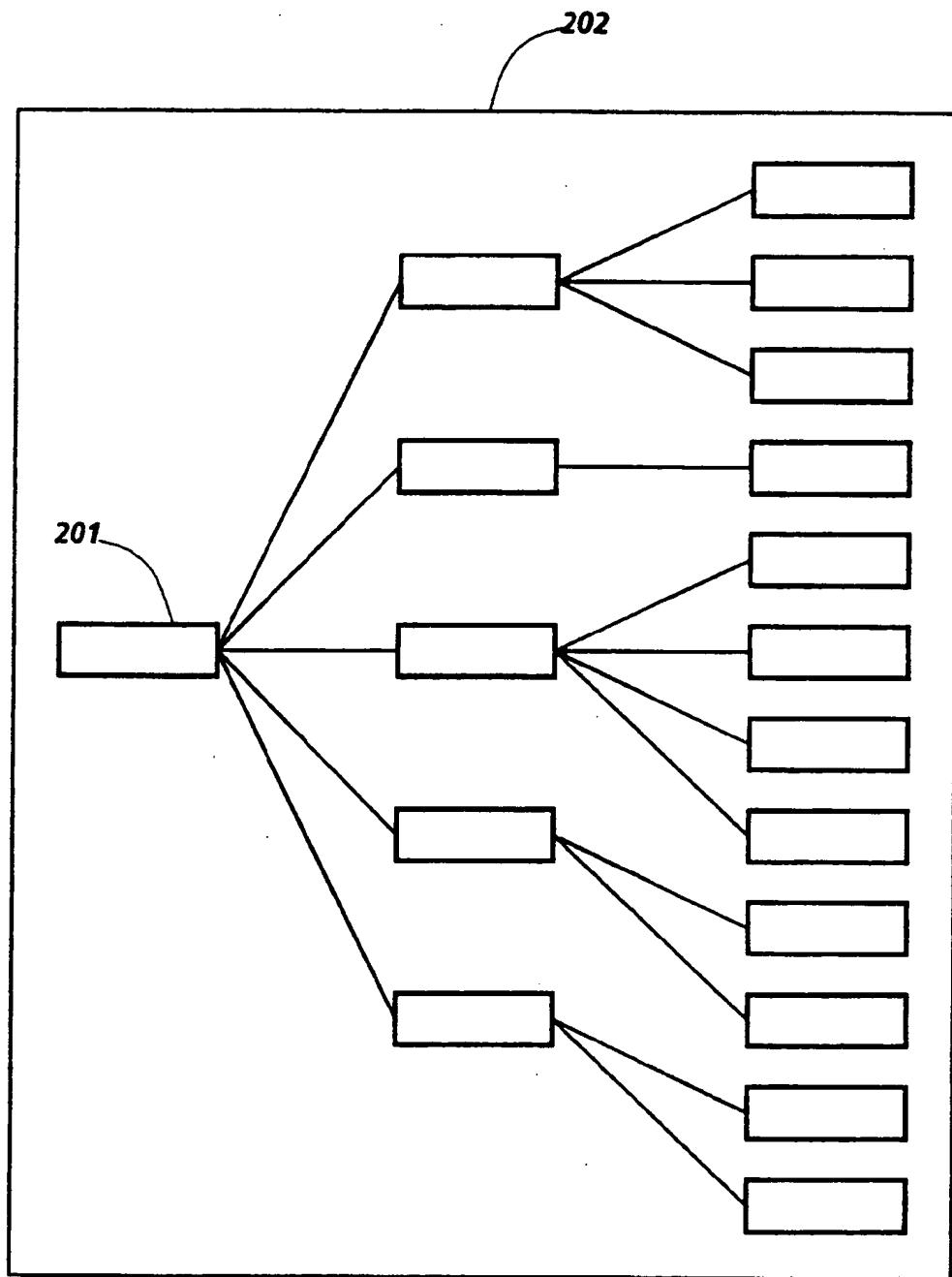


Fig. 2a

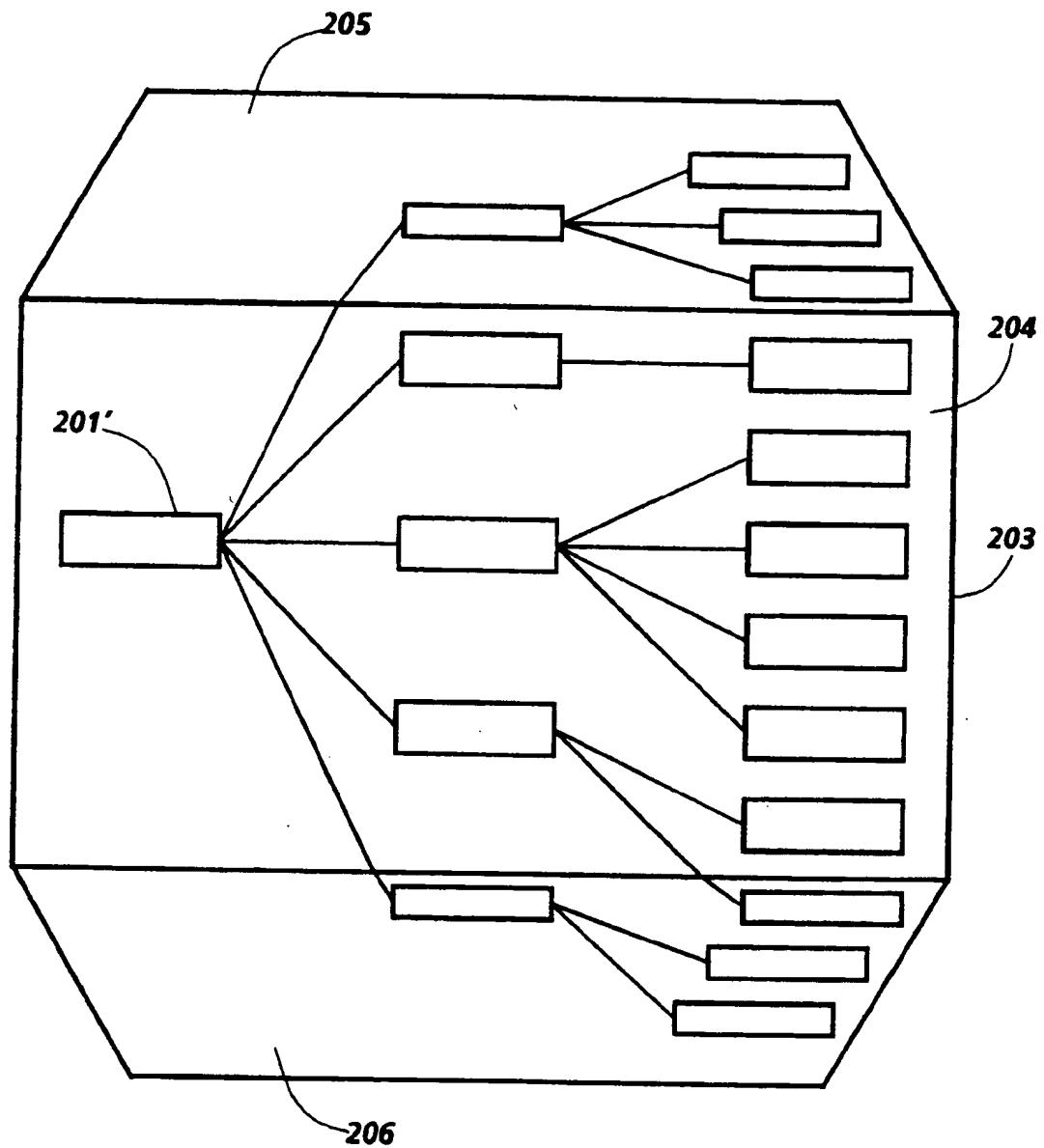


Fig. 2b

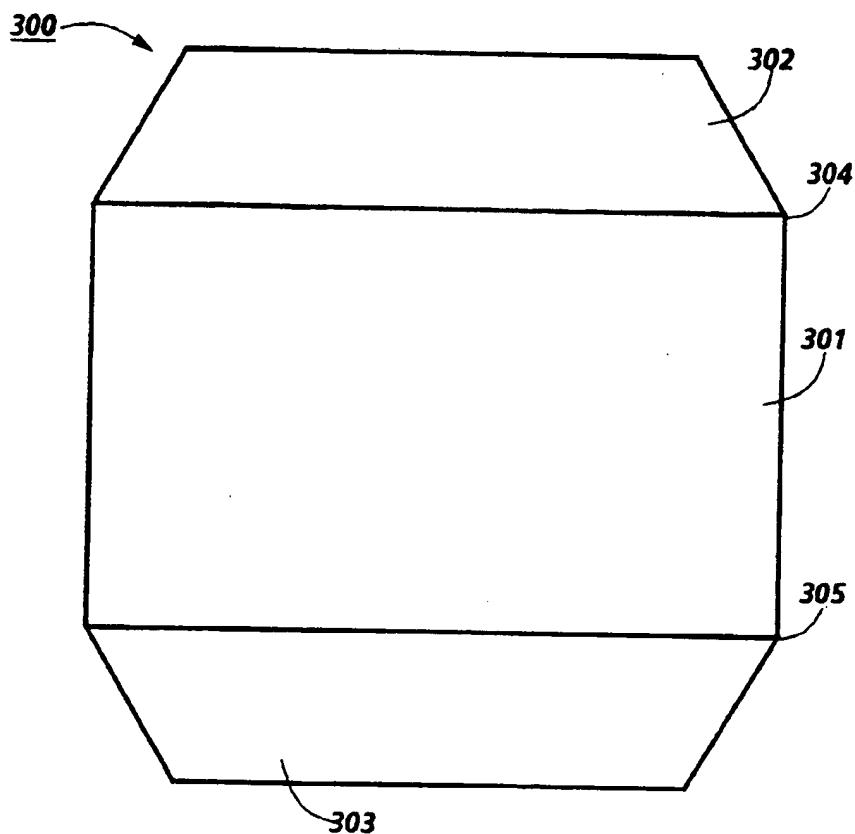


Fig. 3a

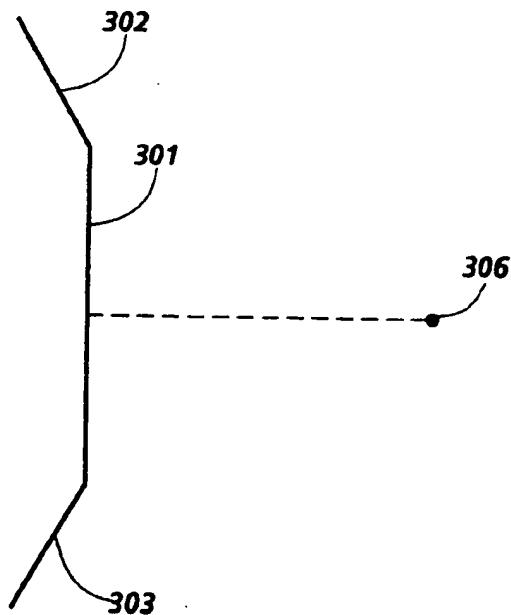


Fig. 3b

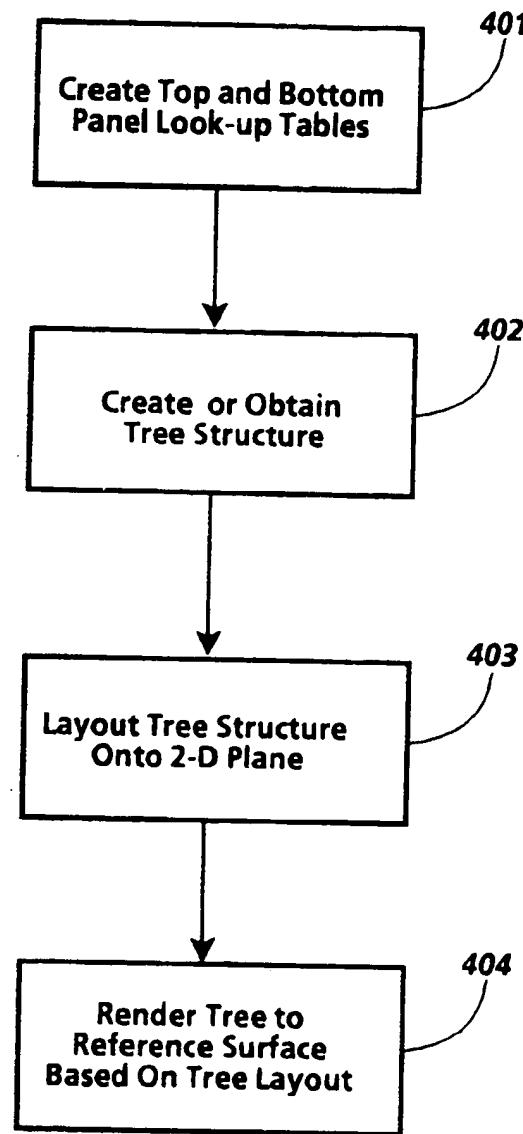


Fig. 4

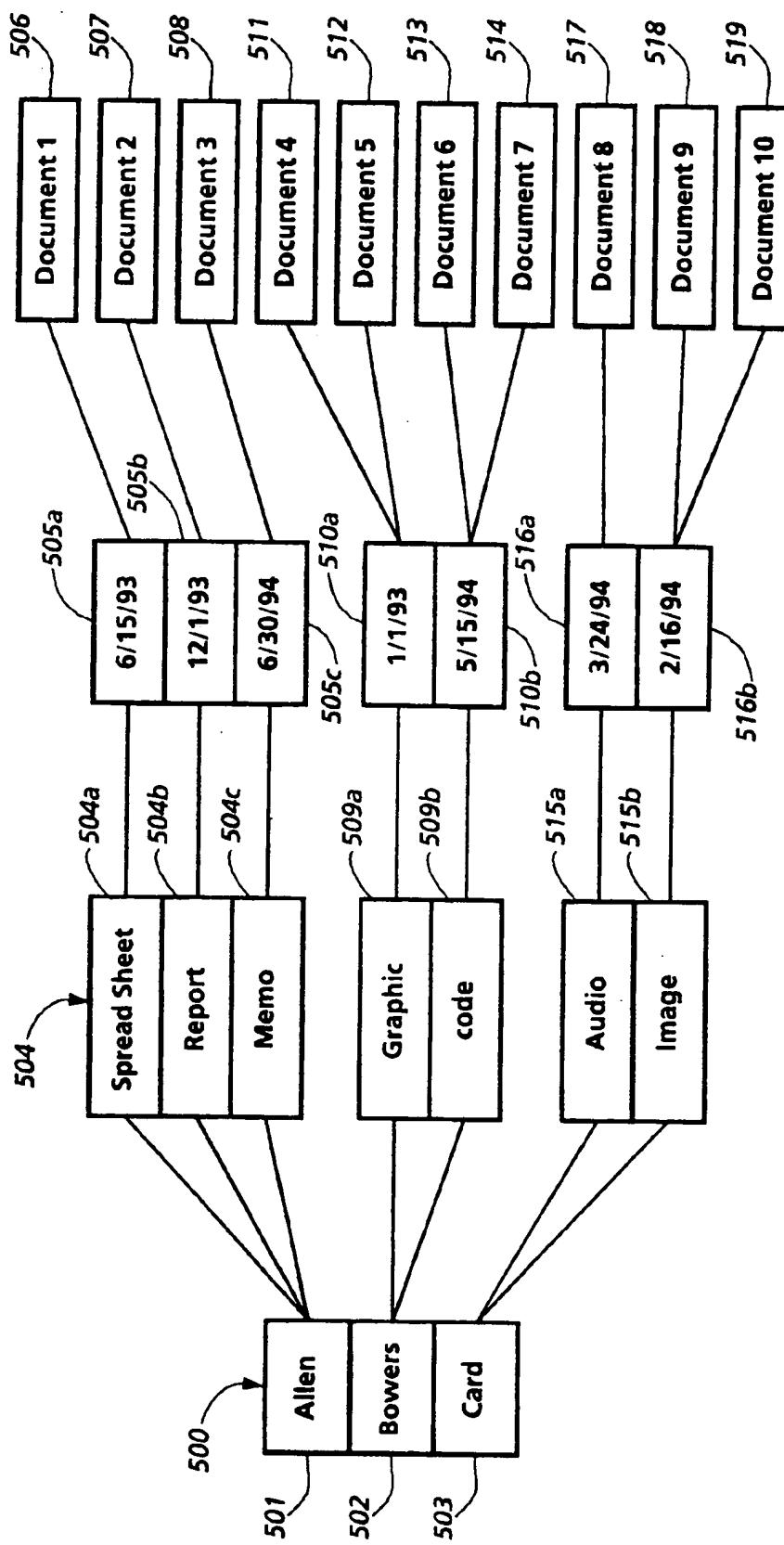


Fig. 5

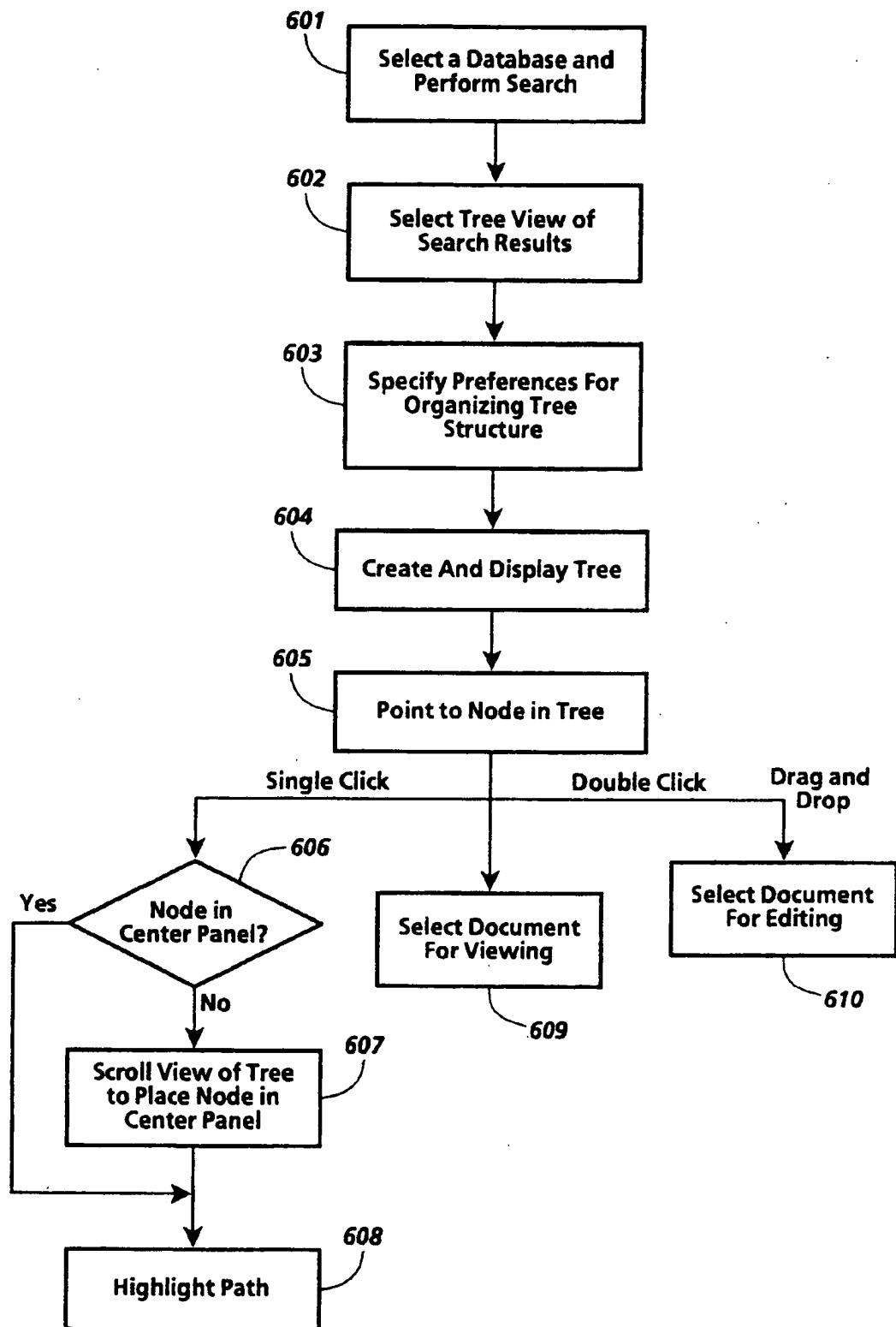


Fig. 6

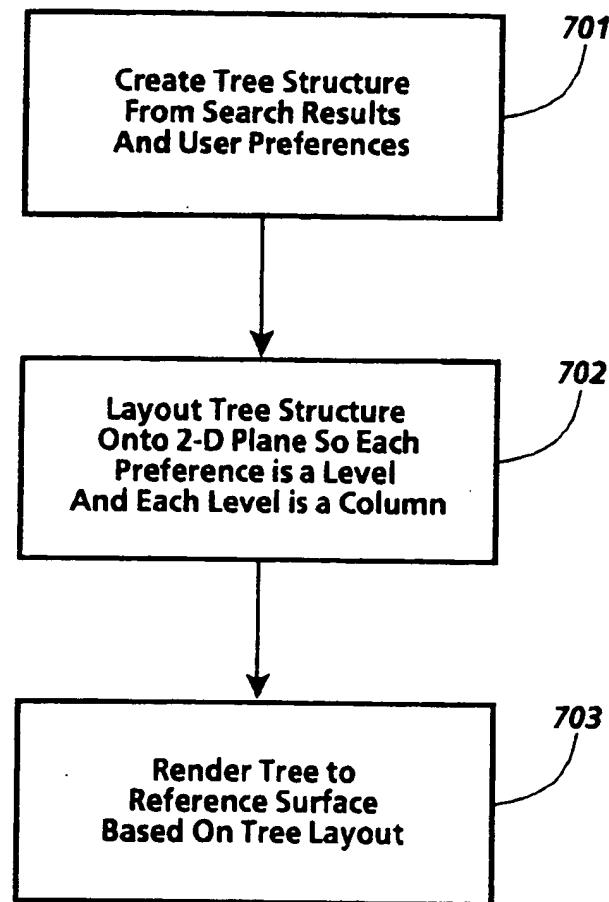
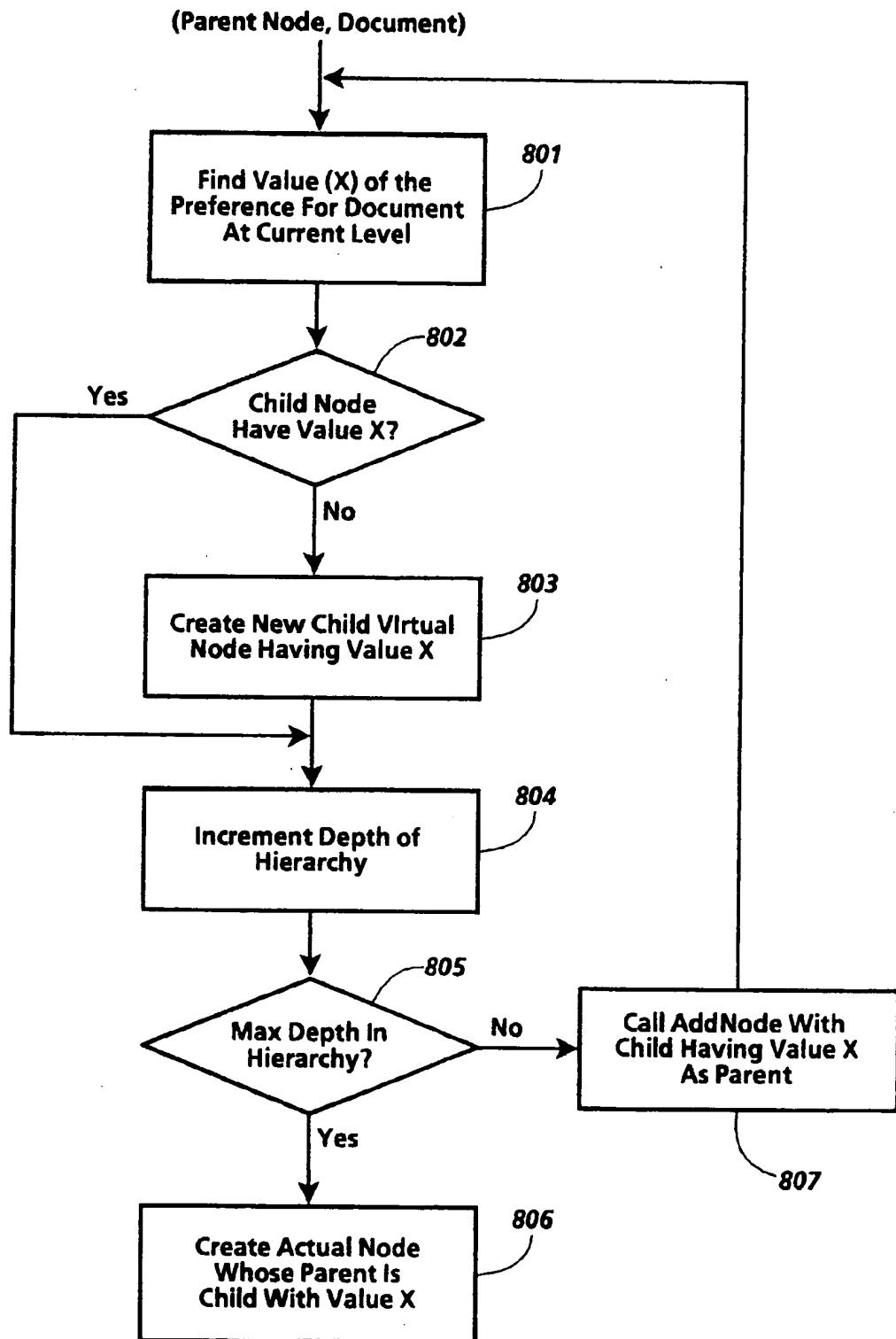


Fig. 7

**Fig. 8**

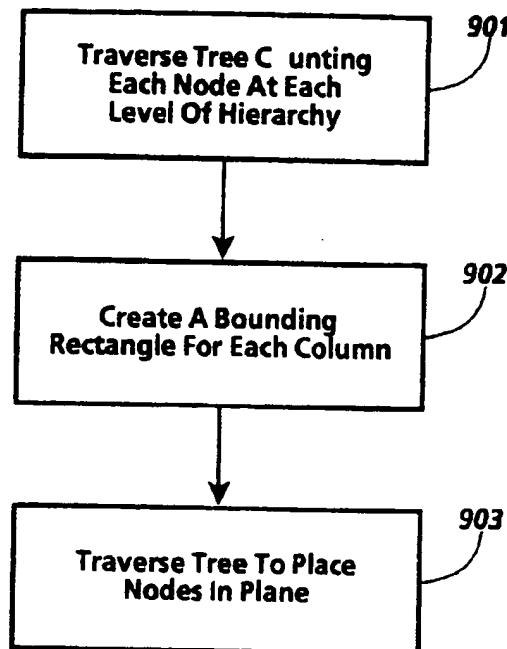


Fig. 9

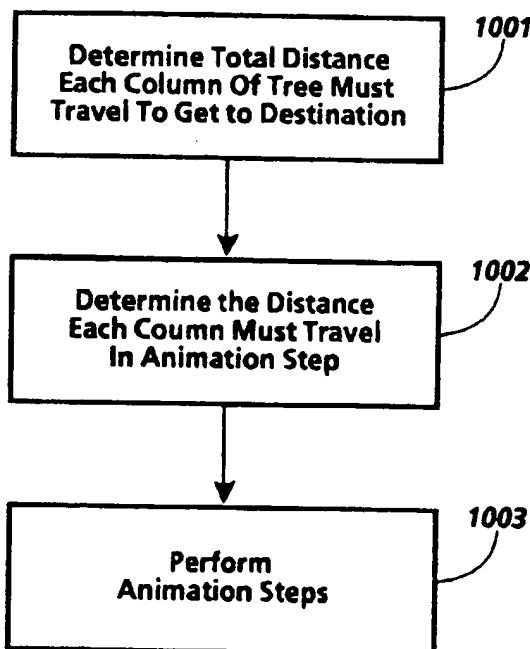


Fig. 10

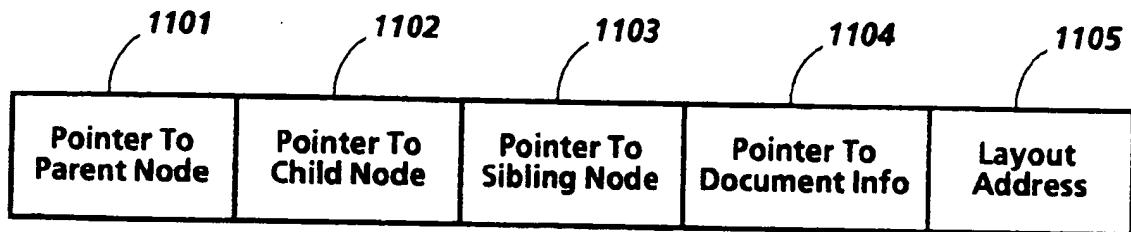


Fig. 11

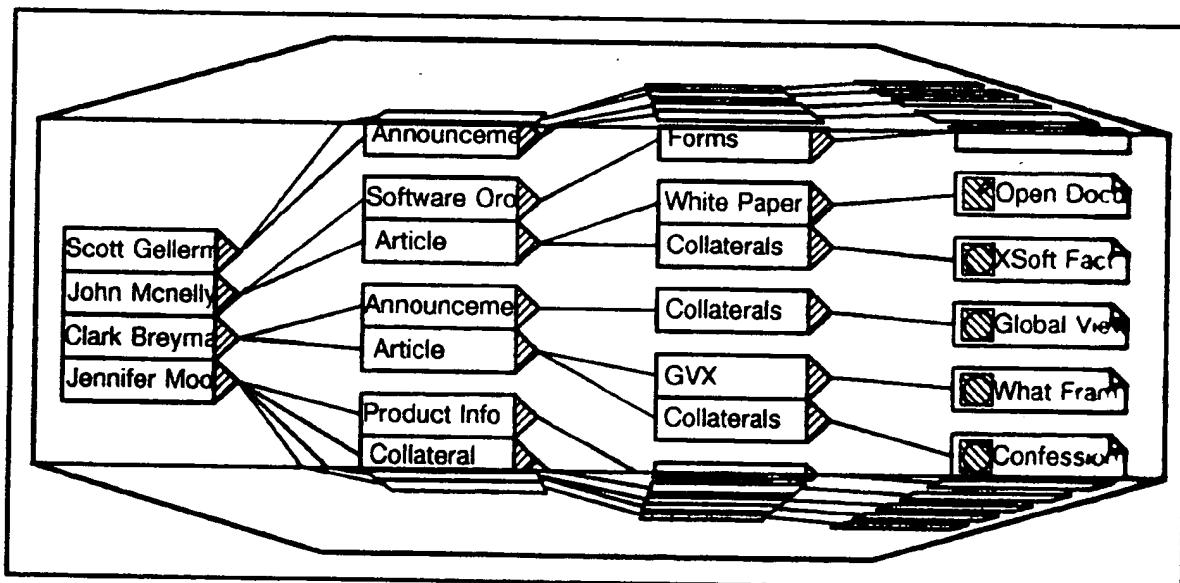
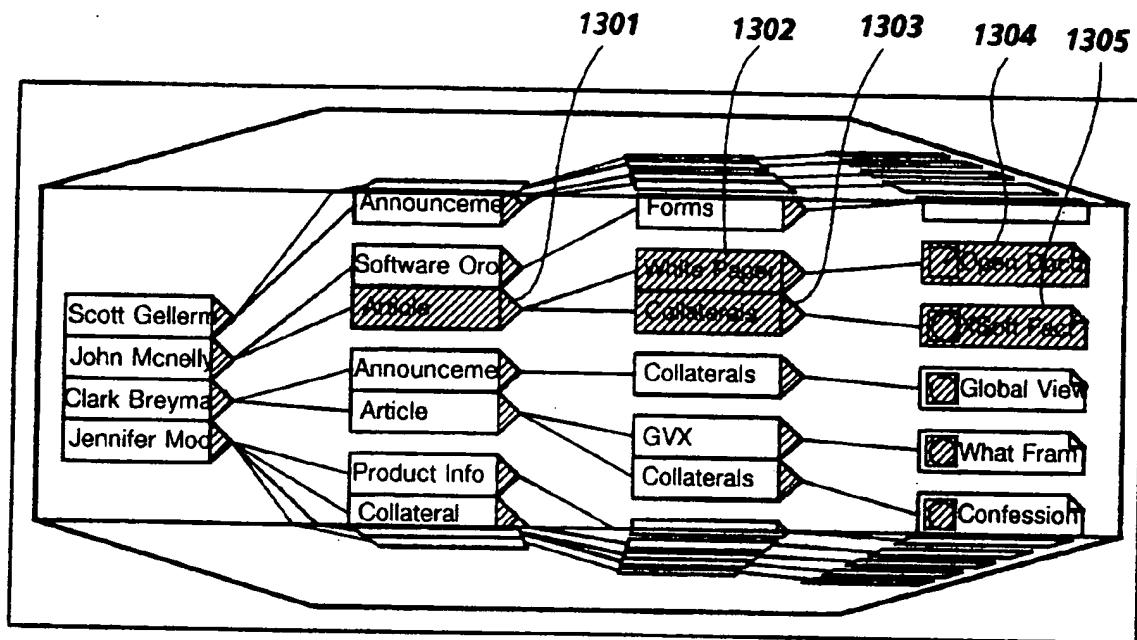
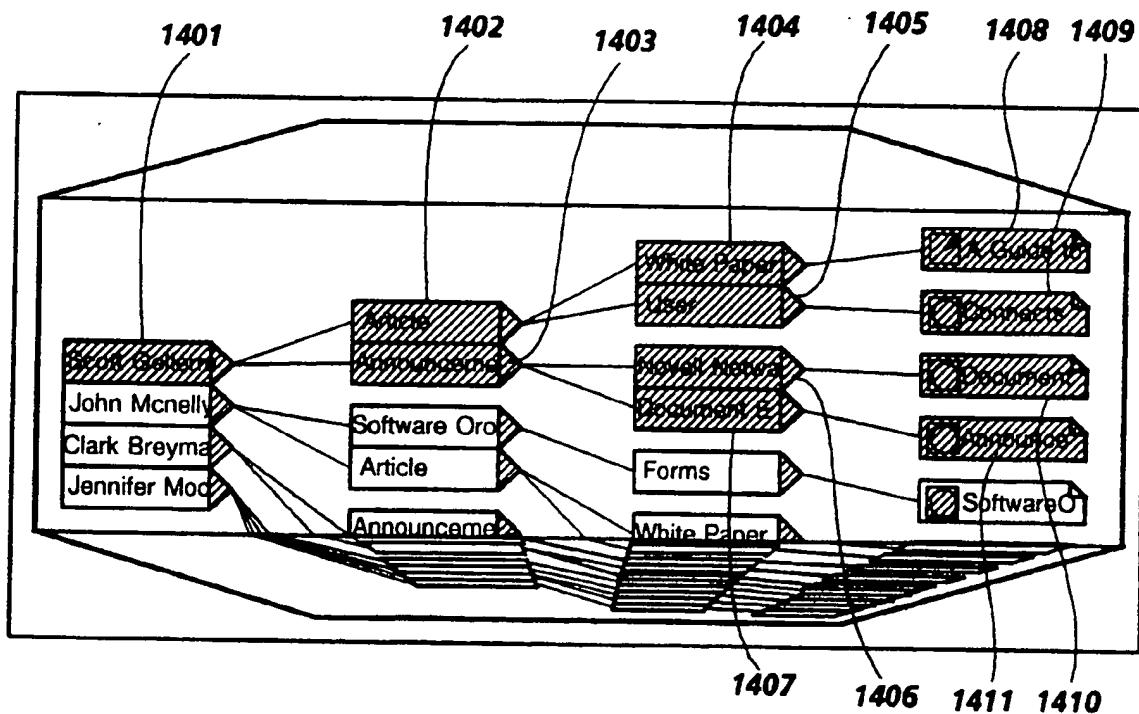


Fig. 12

**Fig. 13****Fig. 14**



EUROPEAN SEARCH REPORT

Application Number
EP 95 30 5052

DOCUMENTS CONSIDERED TO BE RELEVANT			CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	
A	GB-A-2 187 580 (MULTI NET PTY LTD) 9 September 1987 * the whole document * ---	1,5	G06F3/023 G06F17/30
D,A	PROCEEDINGS OF THE CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS III, 17 April 1986 BOSTON, MA, US, pages 16-23, G.W. FURNAS 'Generalized Fisheye Views' * the whole document * ---	1,5	
D,A	EP-A-0 535 986 (XEROX CORP) 7 April 1993 * the whole document * ---	1,5	
D,A	EP-A-0 447 095 (XEROX CORP) 18 September 1991 * the whole document * ---	1,5	
D,A	EP-A-0 435 601 (XEROX CORP) 3 July 1991 * the whole document * ---	1,5	
D,A	US-A-4 752 889 (RAPPAPORT ALAIN T ET AL) 21 June 1988 * the whole document * -----	1,5	TECHNICAL FIELDS SEARCHED (Int.Cl.6) G06F
The present search report has been drawn up for all claims			
Place of search		Date of completion of the search	Examiner
THE HAGUE		25 September 1995	Katerbau, R
CATEGORY OF CITED DOCUMENTS			
X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document			
T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons A : member of the same patent family, corresponding document			